IB897

Intel® Atom™ E3800 SoC Series

3.5" Disk Size SBC

USER'S MANUAL

Version 1.0

Acknowledgments

AMI BIOS is a trademark of American Megatrends Inc.
PS/2 is a trademark of International Business Machines Corporation.
Intel and Atom are registered trademarks of Intel Corporation.
Microsoft Windows is a registered trademark of Microsoft Corporation.
Nuvoton is a registered trademark of Winbond Electronics Corporation.
All other product names or trademarks are properties of their respective owners.

Table of Contents

This page is intentionally left blank.

Introduction

Product Description

IB897 is a 3.5-inch single board computer based on the Intel® Atom™ E3800 series processors. It supports two DDR3L SODIMM sockets for a maximum memory capacity of 8GB.

IB897 features the Intel® Gen7 w/4EUs graphics engines and has both CRT and DisplayPort video display interface, and 24-bit LVDS dual channel interface with the use of the NXP PTN3460 device.

Onboard connections are available for two SATAII ports, two COM ports, one USB 3.0 ports, four USB2.0 ports, audio, two Mini PCI-e(x1) slots, and Micro SD. Power input is made with a +9~+30V DC in interface.

IB897 Features:

- Supports Atom™ E3800 series SoC processors
- Two DDR3L SO-DIMM, 1066/1333 MHz, Max. 8GB memory
- Integrated graphics for VGA, DP displays
- 2 x SATA II connectors
- 2x COM port connectors
- 2 x Mini-PCIe(x1) slot *(w/ USB/mSATA support)*
- 2x GbE (RJ-45) connectors
- 24-bit dual channel LVDS
- Micro SD
- 1x 9V to 30V DC-in power connector

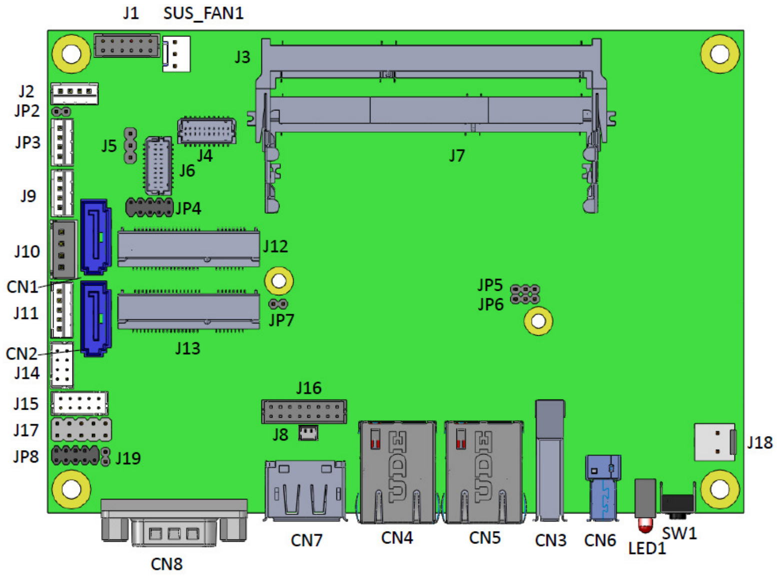
Specifications

Product Name	**IB897 [default silk screen model # on PCB is IB897]**
Form Factor	3.5" disk size SBC
SoC Type/Speed	Intel® Atom™ QC E384**5** /2MB cache/1.91GHz [TDP=10W](IB897-I4**5**) Intel® Atom™ DC E382**7** /1MB cache/1.75 GHz [TDP=8W] (IB897-I2**7**) Package = FCBGA1170,Type-3, 25mmx27mm, 22nm
BIOS	AMI BIOS
Memory	Intel® Atom ™ SoC integrated memory controller **2**x DDR3 SO-DIMM socket [Horizontal type] Max. Memory to DDR3**L**-1333@8GB (Non-ECC, Unbuffered, 1.35V)
VGA	Intel® Gen7 w/4EUs graphics engines (@ 542MHz/792MHz [Turbo]) Supports DX 11, OGL 3.0, OCL 1.1, OGLES 2.0, DP x 1; CRT x 1 via pin header
LVDS	24-bit dual channel via NXP **PTN3460** thru eDP
LAN	Intel I210**I**T PCIe Gigabit LAN x 2 [**9mm x 9mm** @ 64-QFN, -40° C to +85° C]
USB	Intel® Atom™ SoC built-in USB host controller Support USB 2.0 x 4 ports; USB 3.0 x 1 port, USB 2.0 x4 ports (Thru SMSC HUB USB2514)
Serial ATA	Intel® Atom™ SoC built-in SATA II controller, supports 2 ports
Audio	Intel® Atom™ SoC built-in HD Audio controller + Realtek ALC269QHD Codec w/class-D speaker amplifier (2.3W per channel @ 5V power supply) [7mm x 7mm @ 48-QFN]; support 2-channel audio out + amp
LPC I/O	Nuvoton NCT5523D [64-pin LQFP, 7x7x1.4mm] - COM #1 (RS232/422/485) [EXAR SP339EER1 x 1 for jumper-less] - COM #2 (RS-232 only) [Hardware Monitor]: 2x Thermal inputs; 2x Voltage monitoring
Digital IO	4 in & 4 out
Expansion Slots	Mini PCI-e socket x**2** (1xFull-sized+1xHalf-sized,) **Full-sized MiniPCIe (1x) support mSATA**
Edge Connector	DB9 for COM1, DisplayPort, RJ45 x 2 for LAN 1 & 2 USB 2.0 vertical connector x 1 (from SoC) USB 3.0 vertical connector x 1 (from SoC) LED indicators (red+green) x1 for power and HDD status Power button x 1
Onboard Header/ Connector	2x8 pin header for CRT 2x4 pin header for 2x USB 2.0 (from SMSC HUB USB2514) DF20 socket connector x2 for 24-bit dual channel LVDS 4-pin box header for backlight/brightness control (PWM) 2x6 pin box header for Audio, 4-pin header for speaker 2x5 pin box header for COM2 2x5 pin headers for LPC (80-port card debugging purpose) Mini PCI-e(1x) connector x**2**, 5 pins box header for smart battery SATA connector x**2** for SATA device 4-pins power connector (JST type, for SATA device) 2-pins connector for power input, Micro SD slot (type 3.3V)
Watchdog	Yes (256 segments, 0, 1, 2…255 sec/min)
Power Connector	9V ~ 30V DC-in thru onboard 2-pin connector
Others	iSMART 2.0 [Auto-scheduler / Power resume]
OS Support	Windows 8 / Embedded; Windows 7 / Embedded, Linux
RoHS / REACH/ CE /FCC	Yes / Yes / Yes / Class B
Operating Temperature	-40° C to +85° C
Board Size	102mm x 147mm

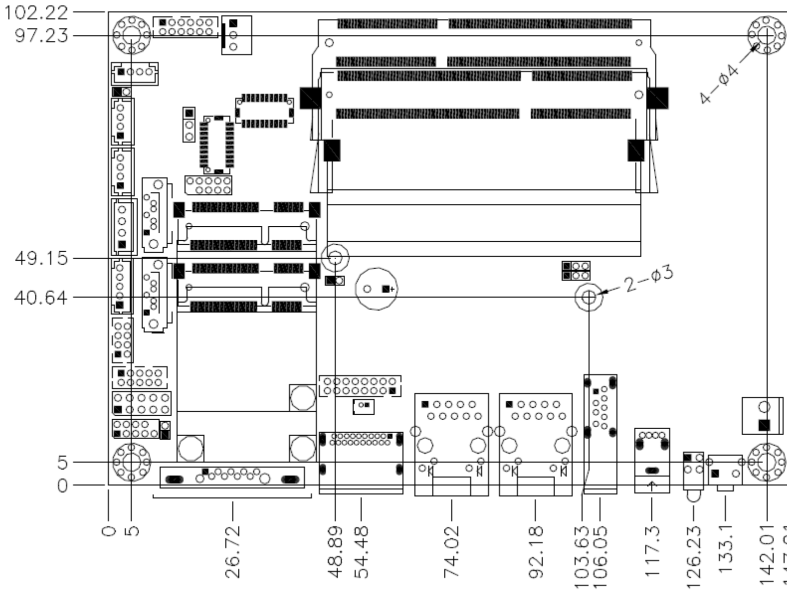
Checklist

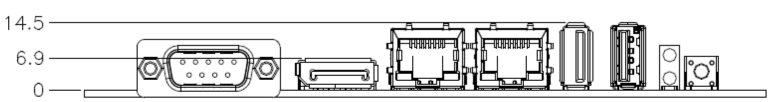
Your IB897 package should include the items listed below.

- The IB897 SBC
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility



Board Dimensions





Installations

This section provides information on how to use the jumpers and connectors on the IB897 in order to set up a workable system. The topics covered are:

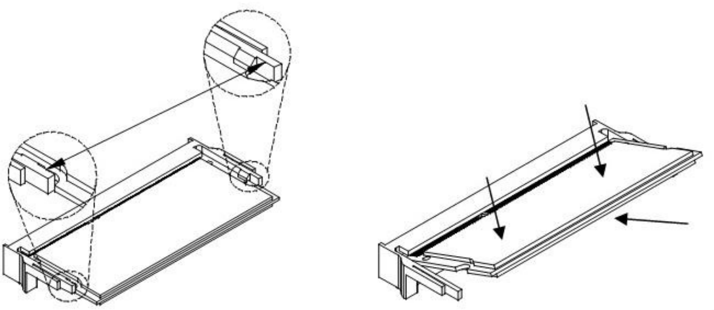
Installing the Memory

The IB897 board supports TWO DDR3L memory socket for a maximum total memory of 8GB DDR3L memory type.

Installing and Removing Memory Modules

To install the DDR3L modules, locate the memory slot on the board and perform the following steps:

1. Hold the DDR3L module so that the key of the DDR3L module aligned with that on the memory slot.
2. Gently push the DDR3L module in an upright position until the clips of the slot close to hold the DDR3L module in place when the DDR3L module touches the bottom of the slot.
3. To remove the DDR3L module, press the clips with both hands.

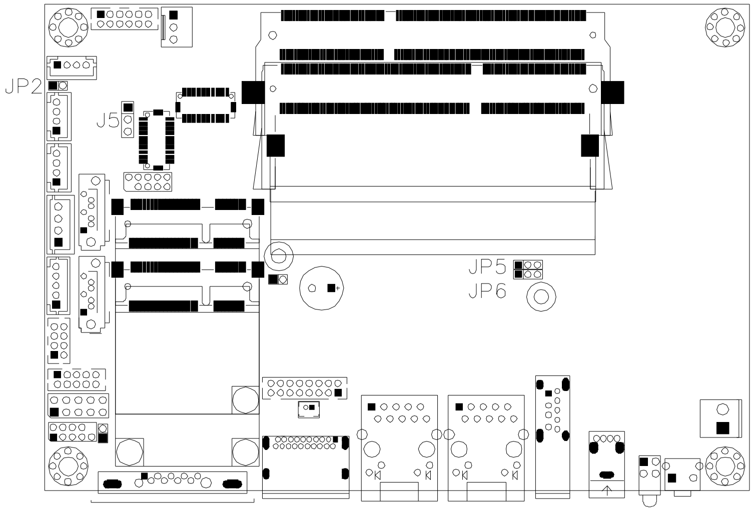


***** Channel-A slot must be installed with memory module for booting up*****

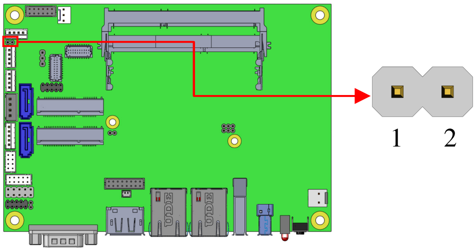
Setting the Jumpers

Jumpers are used on IB897 to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors on IB897 and their respective functions.

Jumper Locations on IB897

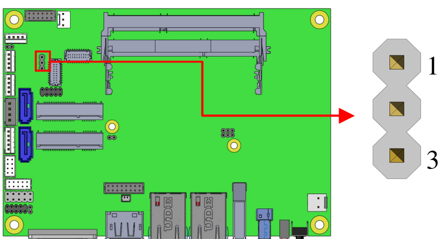


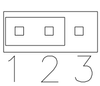
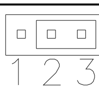
JP2: LVDS Panel Brightness Control Selection



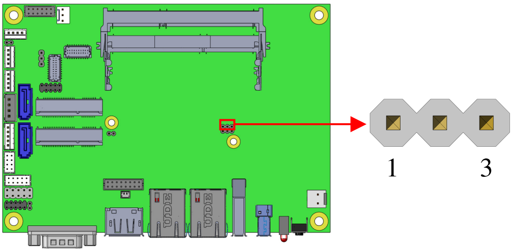
JP2	Brightness Control (PWM mode)
Open	3.3V
Close	5V(Default)

J5: LVDS Panel Power Selection



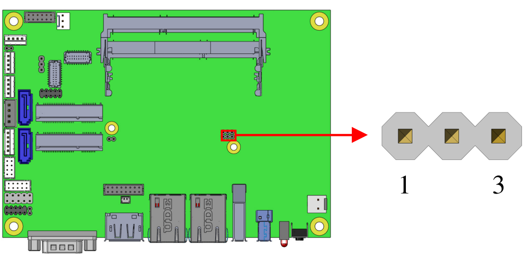
J5	Setting	Panel Voltage
	Pin 1-2 Short/Closed	3.3V (default)
	Pin 2-3 Short/Closed	5V

JP5: Clear ME Contents



JP5	Setting	Function
1 2 3	Pin 1-2 Short/Closed	Normal
1 2 3	Pin 2-3 Short/Closed	Clear ME REGISTER

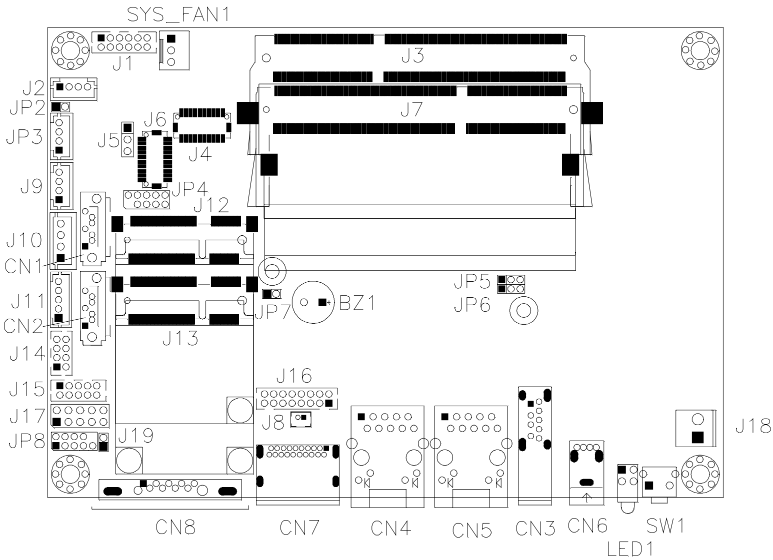
JP6: Clear CMOS Contents

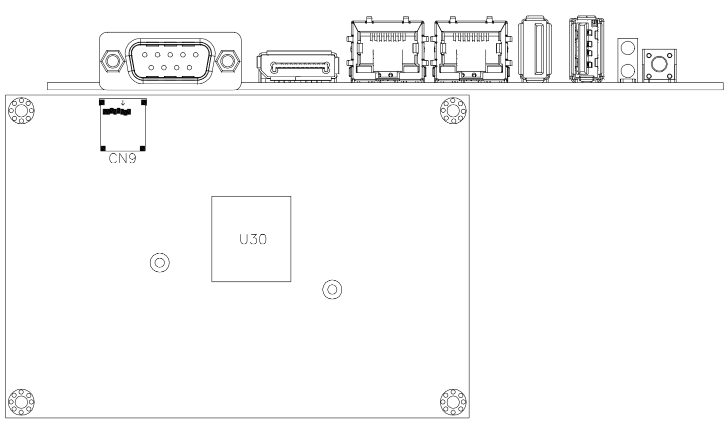


JP6	Setting	Function
1 2 3	Pin 1-2 Short/Closed	Normal
1 2 3	Pin 2-3 Short/Closed	Clear CMOS

Connectors on IB897

Connector Locations on IB897





Bottom side

CN3: USB3.0 Connector

CN4, CN5: Gigabit LAN

CN4: Intel® I210IT

CN5: Intel® I210IT

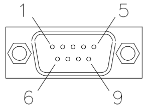
CN6: USB2.0 Connector

CN7: DP Connector

CN8: DB9 Connector (COM1)

Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	6	DSR, Data set ready
RXD, Receive data	2	7	RTS, Request to send
TXD, Transmit data	3	8	CTS, Clear to send
DTR, Data terminal ready	4	9	RI, Ring indicator
GND, ground	5	10	Not Used

COM1 is jumper-less for RS-232, RS-422 and RS-485 and is to be configured with BIOS Selection.



Pin #	Signal Name		
	RS-232	**R2-422**	**RS-485**
1	DCD	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	DTR	RX-	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	RI	NC	NC
10	NC	NC	NC

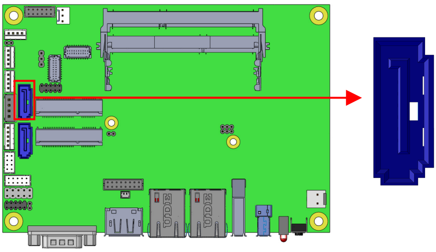
CN9: Micro SD (3.3V) Connector

SW1: Power Switch

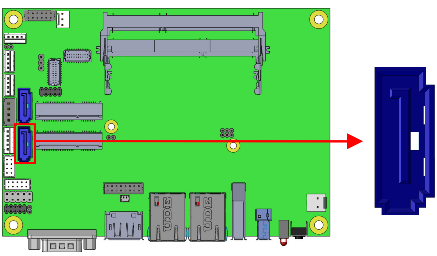
LED1: Power LED and HDD LED

The green LED at the bottom is power LED. The red LED on top is the HDD LED.

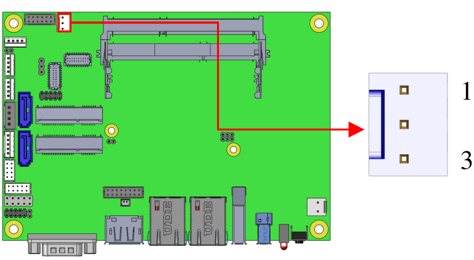
CN1: SATAII /share mSATA/ Connectors



CN2: SATAII Connectors

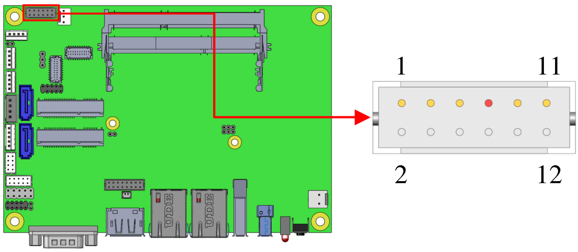


SYS_FAN1: SYSTEM Fan Power Connector



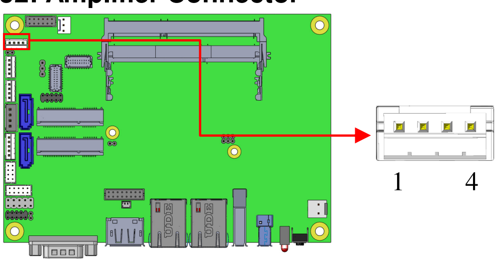
Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection

J1: Audio Connector (DF11-12DP-2DSA)



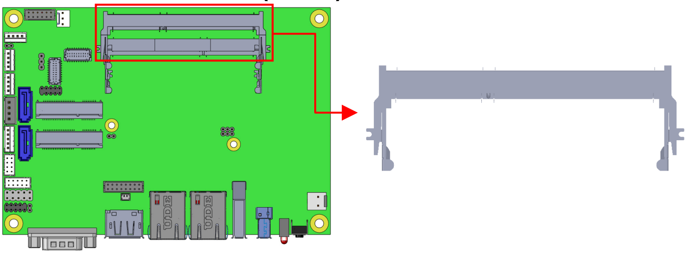
Signal Name	Pin #	Pin #	Signal Name
LINEOUT_R	2	1	LINEOUT_L
Ground	4	3	JD_FRONT
LINEIN_R	6	5	LINEIN_L
Ground	8	7	JD_LINEIN
MIC-R	10	9	MIC_L
Ground	12	11	JD_MIC1

J2: Amplifier Connector



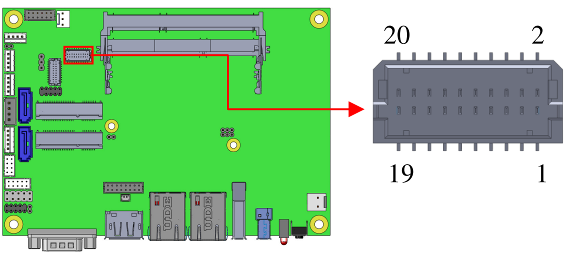
Pin #	Signal Name
1	OUTL+
2	OUTL-
3	OUTR-
4	OUTR+

J3: DDR3L SO-DIMM(CH-B) Sockets

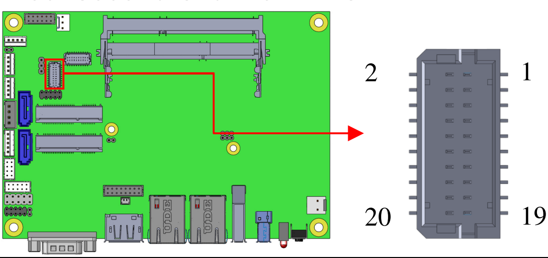


J4, J6: LVDS Connectors, Hirose DF20G-20DP-1V

J4: First Channel LVDS

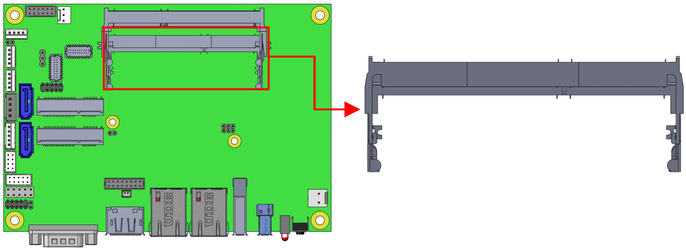


J6: Second Channel LVDS

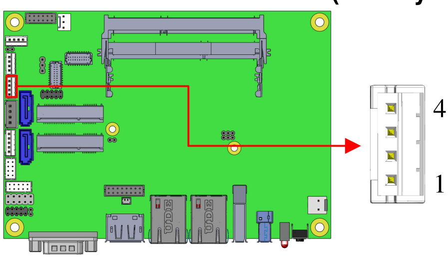


Signal Name	Pin #	Pin #	Signal Name
TX0N	2	1	TX0P
Ground	4	3	Ground
TX1N	6	5	TX1P
Ground	8	7	Ground
TX2N	10	9	TX2P
Ground	12	11	Ground
CLKN	14	13	CLKP
Ground	16	15	Ground
TX3N	18	17	TX3P
Power	20	19	Power

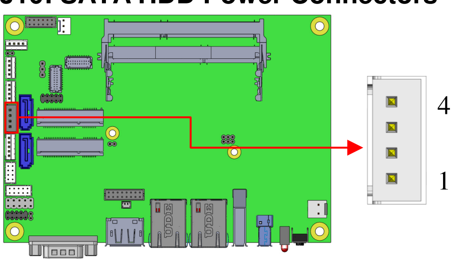
J7: DDR3L SO-DIMM(CH-A) Sockets



J9: MCU Flash Connector (factory use only)

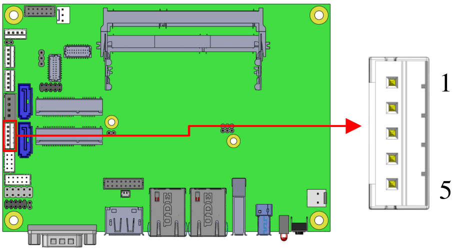


J10: SATA HDD Power Connectors



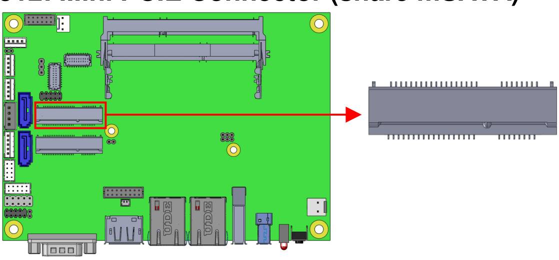
Pin #	Signal Name
1	+5V
2	Ground
3	Ground
4	+12V

J11: Smart Battery

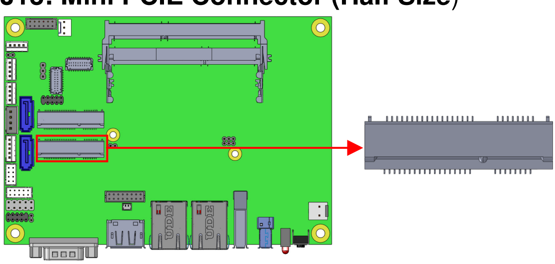


Pin #	Signal Name
1	RST#
2	ICHSWI#
3	Ground
4	SMB_DATA
5	SMB_CLK

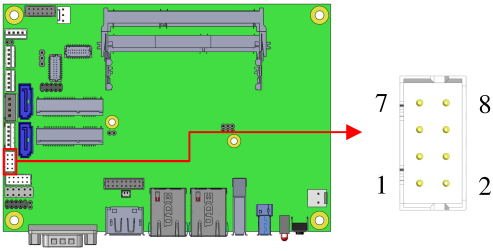
J12: Mini PCIE Connector (share mSATA)



J13: Mini PCIE Connector (Half Size)

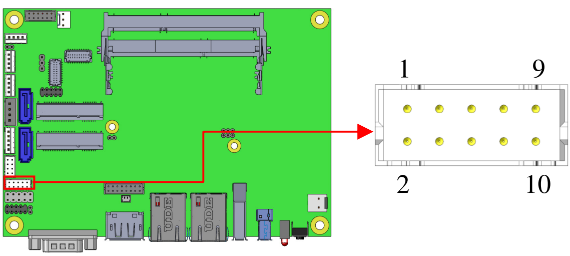


J14: USB 2.0 Connector



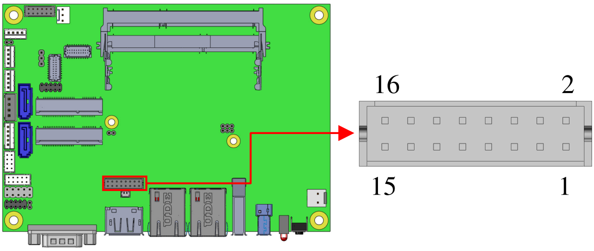
Signal Name	Pin #	Pin #	Signal Name
Vcc	1	2	Ground
D0-	3	4	D1+
D0+	5	6	D1-
Ground	7	8	Vcc

J15: COM2/RS232 Serial Port



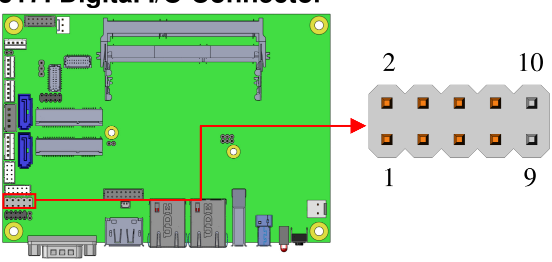
Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	2	RXD, Receive data
TXD, Transmit data	3	4	Data terminal ready
GND, ground	5	6	DSR, Data set ready
RTS, Request to send	7	8	CTS, Clear to send
RI, Ring indicator	9	10	Not Used

J16: VGA Connector (DF11-16DP-2DSA)



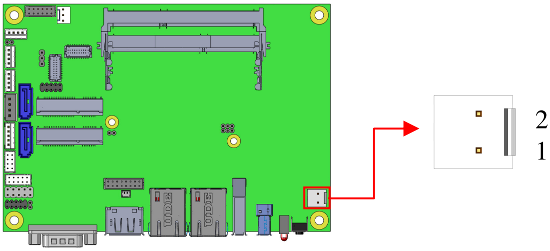
Signal Name	Pin #	Pin #	Signal Name
+5V	2	1	Red
Ground	4	3	Green
N.C	6	5	Blue
DDCDATA	8	7	N.C
H_SYNC	10	9	GND
V_SYNC	12	11	GND
DDCCLK	14	13	GND
N.C.	16	15	GND

J17: Digital I/O Connector



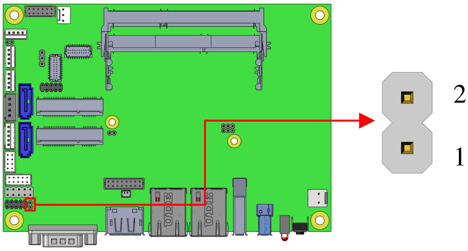
Signal Name	Pin #	Pin #	Signal Name
GND	1	2	VCC
OUT3	3	4	OUT1
OUT2	5	6	OUT0
IN3	7	8	IN1
IN2	9	10	IN0

J18: Board Input Power Connector



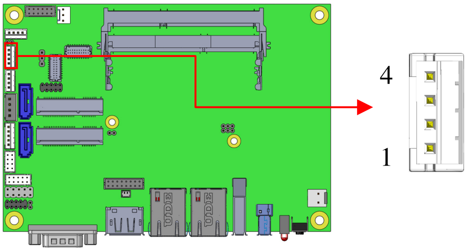
Pin #	Signal Name
1	+9V to +30V
2	GND

J19: Reset Switch



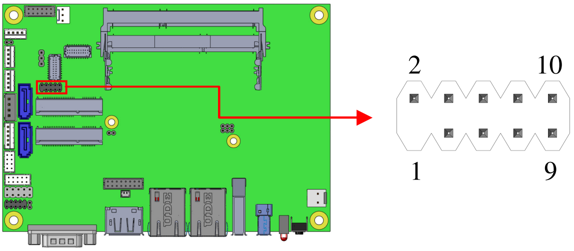
Pin #	Signal Name
1	Reset Switch
2	Ground

JP3: LCD Backlight Connector

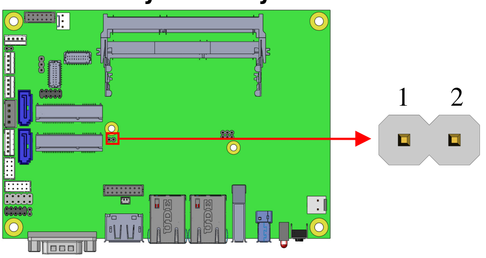


Pin #	Signal Name
1	+12V
2	Backlight Enable
3	Brightness Control
4	Ground

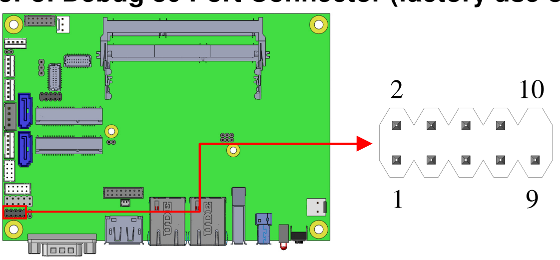
JP4: SPI Flash Connector (factory use only)



JP7: Factory use only



JP8: Debug 80 Port Connector (factory use only)



BIOS Setup

This chapter describes the different settings available in the AMI BIOS that comes with the board. The topics covered in this chapter are as follows:

BIOS Introduction

The BIOS (Basic Input/Output System) installed in your computer system's ROM supports Intel processors. The BIOS provides critical low-level support for a standard device such as disk drives, serial ports and parallel ports. It also password protection as well as special support for detailed fine-tuning of the chipset controlling the entire system.

BIOS Setup

The BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the Setup utility. When you turn on the computer, the BIOS is immediately activated. Pressing the <Del> key immediately allows you to enter the Setup utility. If you are a little bit late pressing the <Del> key, POST (Power On Self Test) will continue with its test routines, thus preventing you from invoking the Setup. If you still wish to enter Setup, restart the system by pressing the "Reset" button or simultaneously pressing the <Ctrl>, <Alt> and <Delete> keys. You can also restart by turning the system Off and back On again. The following message will appear on the screen:

```
Press <DEL> to Enter Setup
```

In general, you press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help and <Esc> to quit.

When you enter the Setup utility, the Main Menu screen will appear on the screen. The Main Menu allows you to select from various setup functions and exit choices.

Warning: *It is strongly recommended that you avoid making any changes to the chipset defaults. These defaults have been carefully chosen by both AMI and your system manufacturer to provide the absolute maximum performance and reliability. Changing the defaults could cause the system to become unstable and crash in some cases.*

Main Settings

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main Advanced Chipset Boot Security Save & Exit		
BIOS Information		Choose the system default language
System Language	[English]	→ ← Select Screen
System Date	[Tue 01/20/2009]	↑ ↓ Select Item Enter: Select +- Change Field F1: General Help
System Time	[21:52:06]	
Access Level	Administrator	F2: Previous Values F3: Optimized Default F4: Save ESC: Exit

System Language

Choose the system default language.

System Date

Set the Date. Use Tab to switch between Data elements.

System Time

Set the Time. Use Tab to switch between Data elements.

Advanced Settings

This section allows you to configure and improve your system and allows you to set up some system features according to your preference.

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main	Advanced	Chipset	Boot	Security	Save & Exit

► ACPI Settings ► LVDS Configuration ► iSmart Controller ► Super IO Configuration ► H/W Monitor ► CPU Configuration ► PPM Configuration ► IDE Configuration ► SDIO Configuration	→←Select Screen ↑↓ Select Item Enter: Select +- Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit

ACPI Settings

Aptio Setup Utility - Copyright © 2013 American Megatrends, Inc.

Main	Advanced	Chipset	Boot	Security	Save & Exit

ACPI Settings Enable ACPI Auto Configuration Enable Hibernation ACPI Sleep State	 Disabled Enabled S3 only (Suspend to …)	→←Select Screen ↑↓ Select Item Enter: Select +- Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit

Enabled ACPI Auto Configuration

Enables or Disables BIOS ACPI Auto Configuration.

Enable Hibernation

Enables or Disables System ability to Hibernate (OS/S4 Sleep State). This option may be not effective with some OS.

ACPI Sleep State

Select ACPI sleep state the system will enter when the SUSPEND button is pressed.

LVDS Configuration

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main **Advanced** Chipset Boot Security Save & Exit		
Configuration		
Panel Color Depth	24 BIT	→←Select Screen
LVDS Channel Type	Single	↑↓ Select Item
Panel Type	1024 x 768	Enter: Select
LVDS Backlight Control	0(Min)	+- Change Opt.
		F1: General Help
		F2: Previous Values
		F3: Optimized Defaults
		F4: Save & Exit
		ESC: Exit

iSmart Controller

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main **Advanced** Chipset Boot Security Save & Exit		
iSmart Controller		→←Select Screen
Power-On after Power failure	Disable	↑↓ Select Item
Schedule Slot 1	None	Enter: Select
Schedule Slot 2	None	+- Change Opt.
		F1: General Help
		F2: Previous Values
		F3: Optimized Defaults
		F4: Save & Exit
		ESC: Exit

Power-On after Power failure

This field sets the system power status whether *Disable or Enable* when power returns to the system from a power failure situation.

Schedule Slot 1 / 2

Setup the hour/minute for system power on.

Super IO Configuration

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main Advanced Chipset Boot Security Save & Exit	
Super IO Configuration ► Serial Port 1 Configuration ► Serial Port 2 Configuration	→←Select Screen ↑↓ Select Item Enter: Select +- Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit

Serial Port 1 Configuration
Set parameters of serial port 1(COMA)

Serial Port 2 Configuration
Set parameters of serial port 2(COMA)

H/W Monitor

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main	Advanced	Chipset	Boot	Security	Save & Exit

PC Health Status		
Smart Fan Function	Disabled	
SYS temp	+33.0 C	
CPU temp	+34.5 C	→←Select Screen
FAN1 Speed	4066 RPM	↑↓ Select Item
Vcore	+1.704 V	Enter: Select
+1.35V	+1.544 V	+- Change Opt.
AVCC	+3.360 V	F1: General Help
VSB3	+3.344 V	F2: Previous Values
VCC3V	+3.328 V	F3: Optimized Defaults
		F4: Save & Exit
CPU Shutdown Temperature	Disabled	ESC: Exit

Smart Fan Function

This field enables or disables the smart fan feature.
Disabled (default)
50 ℃
60 ℃
70 ℃
80 ℃
90 ℃

Shutdown Temperature

This field enables or disables the Shutdown Temperature
Disabled (default)
70 ℃/158 F
75 ℃/167 F
80 ℃/176 F
85 ℃/185 F
90 ℃/194 F
90 ℃/203 F

Temperatures/Voltages

These fields are the parameters of the hardware monitoring function feature of the motherboard. The values are read-only values as monitored by the system and show the PC health status

CPU Configuration

This section shows the CPU configuration parameters.

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main **Advanced** Chipset Boot Security Save & Exit	
CPU Configuration	
►Socket 0 CPU Information	
CPU Speed 1751 Mhz	
64-bit Supported	
	→←Select Screen ↑↓ Select Item Enter: Select +- Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit

Socket 0 CPU Information

Socket specific CPU Information.

CPU PPM Configuration

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main	Advanced	Chipset	Boot	Security	Save & Exit
CPU PPM Configuration					
EIST			Enabled		→ ←Select Screen ↑ ↓ Select Item Enter: Select +- Change Field F1: General Help F2: Previous Values F3: Optimized Default F4: Save ESC: Exit

EIST
Enable/Disable Intel SpeedStep.

IDE Configuration

SATA Devices Configuration.

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main **Advanced** Chipset Boot Security Save & Exit	
IDE Configuration	
Serial-ATA (SATA) Enabled	
SATA Mode AHCI	
Serial-ATA Port 0 Enabled	
SATA Port0 HotPlug Disabled	
Serial-ATA Port 1 Enabled	
SATA Port1 HotPlug Disabled	
SATA Port0	→←Select Screen
Not Present	↑↓ Select Item
	Enter: Select
SATA Port1	+- Change Field
Not Present	F1: General Help
	F2: Previous Values
	F3: Optimized Default
	F4: Save ESC: Exit

Serial-ATA(SATA)

Enabled / Disabled Serial ATA

SATA Mode

Select IDE / AHCI Mode

Serail –ATA Port 0

Enabled / Disabled Serial Port 0

SATA Port0 HotPlug

Enabled / Disabled SATA Port 0 HotPlug

Serail –ATA Port 1

Enabled / Disabled Serial Port 1

SATA Port1 HotPlug

Enabled / Disabled SATA Port 1 HotPlug

SDIO Configuration

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main **Advanced** Chipset Boot Security Save & Exit	
SDIO Access Mode Auto	→ ←Select Screen ↑ ↓ Select Item Enter: Select +- Change Field F1: General Help F2: Previous Values F3: Optimized Default F4: Save ESC: Exit

SDIO Access Mode

Auto Option: Access SD device in DMA mode if controller supports it. Otherwise, use PIO mode.

DMA options: Access SD device in DMA mode.

PIO Option: Access PIO device in DMA

Chipset Settings

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main	Advanced	**Chipset**	Boot	Security	Save & Exit

► North Bridge	→←Select Screen ↑↓ Select Item Enter: Select +- Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit

North Bridge

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main	Advanced	**Chipset**	Boot	Security	Save & Exit

Memory Information		→← Select Screen ↑↓ Select Item Enter: Select +- Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit
Total Memory	4096 MB (LPDDR3)	
Memory Slot0	4096 MB (LPDDR3)	
Memory Slot2	Not Present	

Security Settings

This section allows you to configure and improve your system and allows you to set up some system features according to your preference.

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main	Advanced	Chipset	Boot	**Security**	Save & Exit

Password Description If ONLY the Administrator's password is set, then this only limit access to Setup and is only asked for when entering Setup. If ONLY the User's password is set, then this is a power on password and must be entered to boot or enter Setup. In Setup the User will have Administrator rights The password length must be in the following range:		→ ←Select Screen ↑ ↓ Select Item Enter: Select +- Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit
Minimum length	3	
Maximum length	20	
Administrator Password		
User Password		

Administrator Password

Set Administrator Password.

Boot Settings

This section allows you to configure the boot settings.

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main Advanced Chipset **Boot** Security Save & Exit		
Boot Configuration		
Setup Prompt Timeout	1	
Bootup NumLock State	On	
		→←Select Screen
Quiet Boot	Disabled	↑↓ Select Item
Fast Boot	Disabled	Enter: Select
		+- Change Opt.
Boot Option Priorities		F1: General Help
Boot Option #1	UEFI:Built-in EFI	F2: Previous Values
		F3: Optimized Defaults
		F4: Save & Exit
		ESC: Exit

Setup Prompt Timeout

Number of seconds to wait for setup activation key.
65535(0xFFFF) means indefinite waiting.

Bootup NumLock State

Select the keyboard NumLock state.

Quiet Boot

Enables or disables Quiet Boot option.

Fast Boot

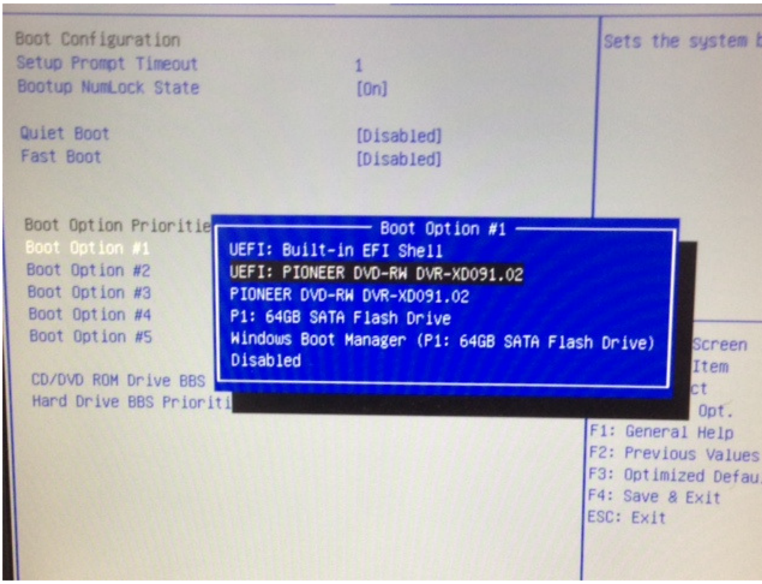
Enables or disables boot with initialization of a minimal set of devices required to launch active boot option. Has no effect for BBS boot options.

Boot Option Priorities

Sets the system boot order.

REMARKS:

Before the installation of Windows 8, go to the BIOS Setup to check that the Boot Option #1 of the Boot Option Priorities field is set as UEFI MODE (DVD drive).



Save & Exit Settings

Aptio Setup Utility – Copyright © 2013 American Megatrends, Inc.

Main Advanced Chipset Boot Security **Save & Exit**	
Save Changes and Exit	
Discard Changes and Exit	
Save Changes and Reset	
Discard Changes and Reset	
Save Options	→←Select Screen
Save Changes	↑↓ Select Item
Discard Changes	Enter: Select
	+- Change Opt.
Restore Defaults	F1: General Help
Save as User Defaults	F2: Previous Values
Restore User Defaults	F3: Optimized Defaults
	F4: Save & Exit
Boot Override	ESC: Exit

Save Changes and Exit
Exit system setup after saving the changes.

Discard Changes and Exit
Exit system setup without saving any changes.

Save Changes and Reset
Reset the system after saving the changes.

Discard Changes and Reset
Reset system setup without saving any changes.

Save Changes
Save Changes done so far to any of the setup options.

Discard Changes
Discard Changes done so far to any of the setup options.

Restore Defaults
Restore/Load Defaults values for all the setup options.

Save as User Defaults
Save the changes done so far as User Defaults.

Restore User Defaults
Restore the User Defaults to all the setup options.

Drivers Installation

This section describes the installation procedures for software and drivers. The software and drivers are included with the motherboard. If you find the items missing, please contact the vendor where you made the purchase. The contents of this section include the following:

IMPORTANT NOTE:
After installing your Windows operating system, you must install first the Intel Chipset Software Installation Utility before proceeding with the drivers installation.

Intel Chipset Software Installation Utility

The Intel Chipset Drivers should be installed first before the software drivers to enable Plug & Play INF support for Intel chipset components. Follow the instructions below to complete the installation.

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) Baytrail Chipset***. Click ***Intel(R) Chipset Software Installation Utility.***





3. When the Welcome screen to the Intel® Chipset Device Software appears, click ***Next*** to continue.

4. Click ***Yes*** to accept the software license agreement and proceed with the installation process.

5. The Setup process is now complete. Click ***Finish*** to restart the computer and for changes to take effect.

VGA Drivers Installation

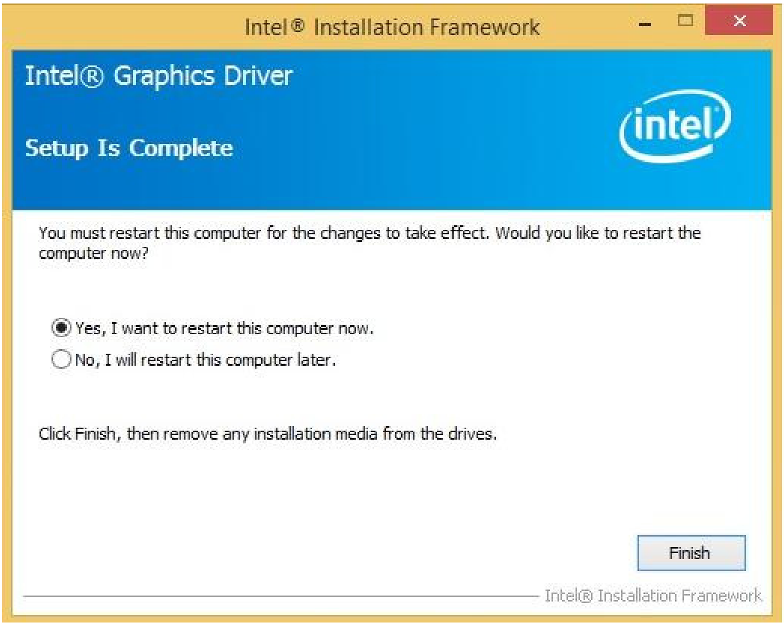
1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) Baytrail Chipset***. Click ***Intel(R) Baytrail Graphics Driver.***



2. When the Welcome screen appears, click ***Next*** to continue.

3. Click ***Yes*** to accept the license agreement and continue the installation.

4. Setup complete. Click ***Finish*** to restart the computer and for changes to take effect.

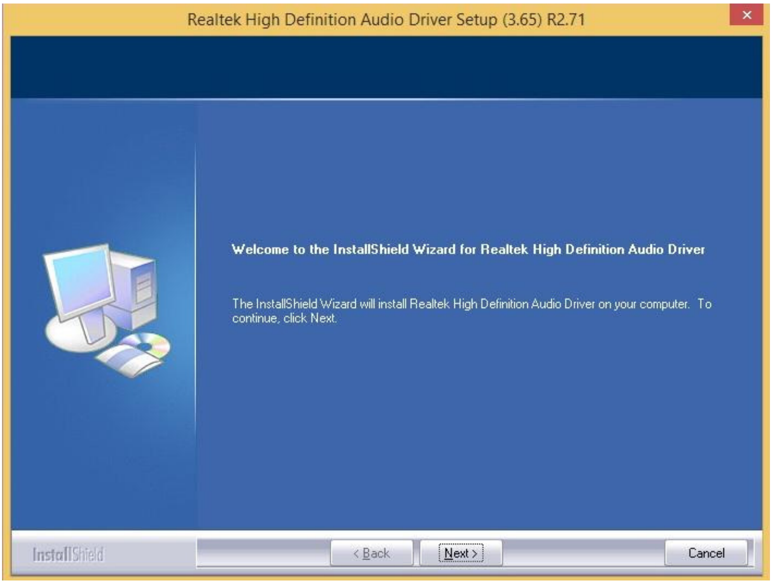


Realtek High Definition Audio Driver Installation

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) Baytrail Chipset***. Click ***Realtek High Definition Audio Driver.***



2. On the Welcome screen, click ***Next*** to proceed with the installation.



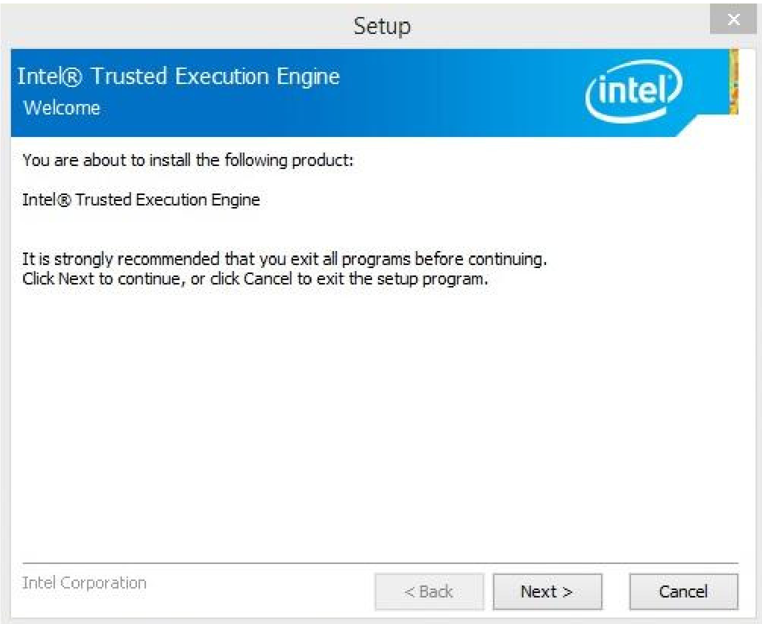
4. InstallShield Wizard is complete. Click ***Finish*** to restart the computer and for changes to take effect.

Intel Trusted Execution Engine Installation

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) Baytrail Chipset***. Click ***Intel(R) Baytrail Graphics Driver.***



2. On the Setup Welcome screen, click ***Next*** to proceed with the installation process.



3. Click ***Next*** accept the license agreement and continue the installation.

4. Installation of the Intel Trusted Execution Engine is now complete. Click ***Finish***.

Intel Sideband Fabric Device Driver Installation

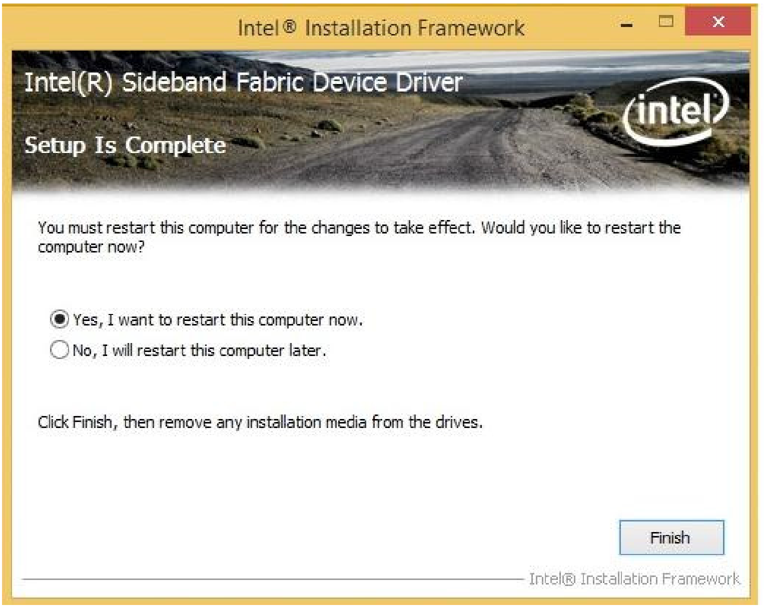
1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) Baytrail Chipset***. Click ***Intel(R) MBI Drivers.***



2. On the Welcome to the Setup screen, click ***Next*** to proceed.

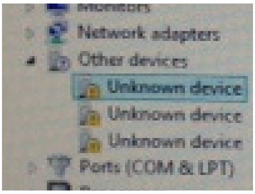
3. Click ***Next*** to accept the license agreement and to proceed.

4. Setup is complete. Click ***Finish*** to restart the computer and for changes to take effect.

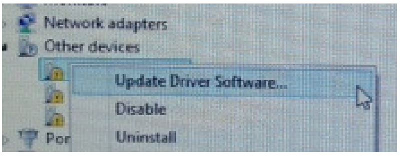


GPIO Controller Driver Installation

1. The driver for the GPIO controllers has to be manually installed. Go to the Device Manager. There will be an item for **Other devices** showing several **Unknown device.**



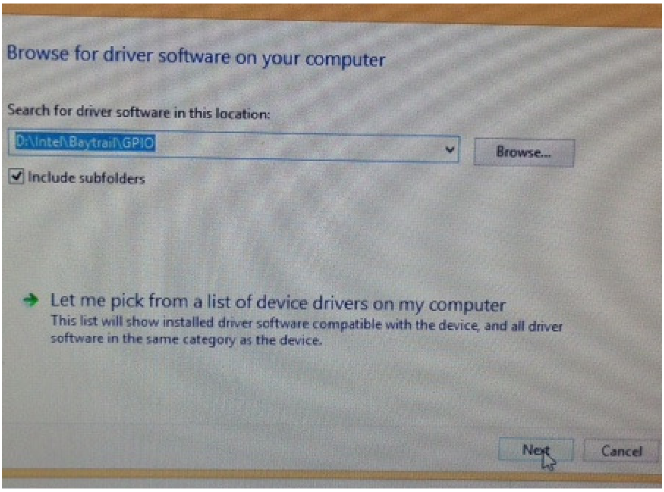
2. With the mouse, right click on the first **Unknown device** and select **Update Driver Software.**



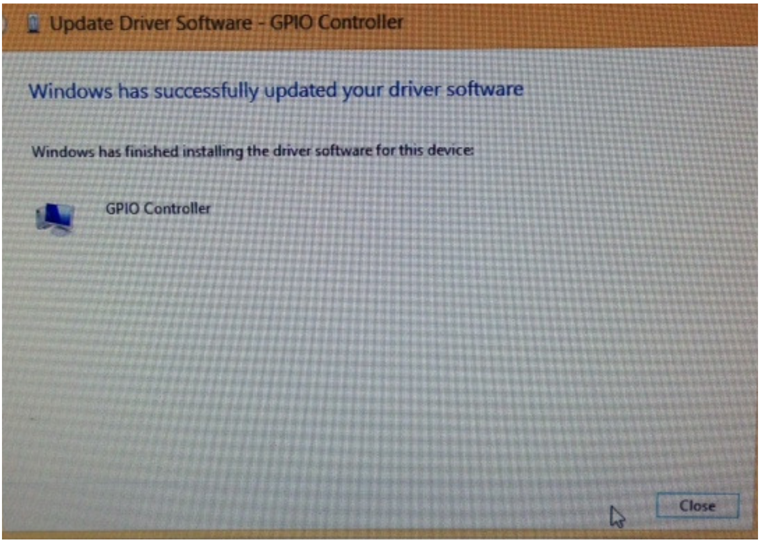
3. Click **Browse** and select the folder that contains the drivers, which is the drive for the drivers DVD. As shown below, click on GPIO, then click **OK**.



4. When the driver software in the location has been found, click **Next** to proceed with the installation.

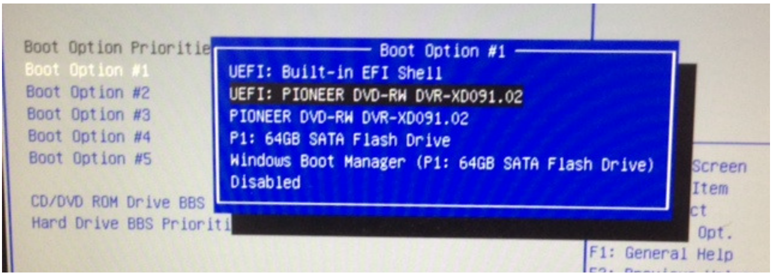


3. Windows has successfully updated the driver software for the GPIO controller. Click **Close**. Repeat the same steps for the remaining two GPIO controllers.

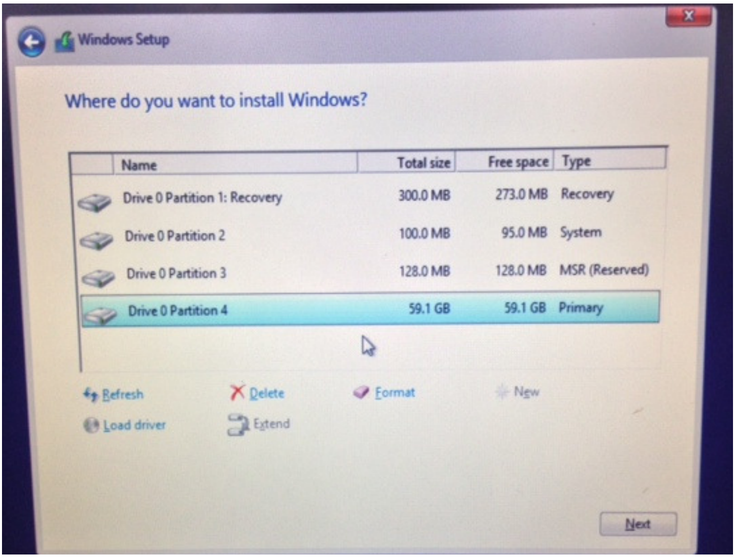


Notes on Boot Option and Windows 8 Setup

1. Before installing Windows 8, check the BIOS Setup's **Boot Option Priorities** if **Boot Option #1** is set to UEFI mode (which would be the DVD drive where to boot the Windows 8 OS from). UEFI stands for Unified Extensible Firmware Interface, an interface between the operating systems and platform firmware during the start-up process.



2. When Boot Option #1 is set to UEFI, there will be 4 partitions in the HDD shown in the Windows Setup screen as can be seen in the picture below.



Appendix

A. I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses which also becomes the identity of the device. The following table lists the I/O port addresses used.

Address	Device Description
0000h-001Fh	Direct memory access controller
0000h-001Fh	PCI bus
0040h-0043h	System timer
0050h-0053h	System timer
0070h-0077h	System CMOS/real time clock
0081h-0091h	Direct memory access controller
0093h-009Fh	Direct memory access controller
00C0h-00DFh	Direct memory access controller
00F0h-00F0h	Numeric data processor
02F8h-02FFh	Communications Port (COM2)
03B0h-03BBh	Intel(R) HD Graphics 4600
03C0h-03DFh	Intel(R) HD Graphics 4600
03F8h-03FFh	Communications Port (COM1)
0D00h-FFFFh	PCI bus
E000h-EFFFh	Intel(R) 8 Series/C220 Series PCI Express Root Port #7 - 8C1C
F000h-F03Fh	Intel(R) HD Graphics 4600
F040h-F05Fh	Intel(R) 8 Series/C220 Series SMBus Controller - 8C22
F060h-F07Fh	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0A0h-F0A3h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0B0h-F0B7h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0C0h-F0C3h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0D0h-F0D7h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0E0h-F0E7h	Intel(R) Active Management Technology - SOL (COM3)

B. Interrupt Request Lines (IRQ)

Peripheral devices use interrupt request lines to notify CPU for the service required. The following table shows the IRQ used by the devices on board.

Level	Function
IRQ0	System Timer
IRQ3	Serial Port #2
IRQ4	Serial Port #1
IRQ 10	Intel(R) 8 Series/C220 Series SMBus Controller - 8C22
IRQ 13	Numeric data processor
IRQ 16	High Definition Audio Controller
IRQ 16	Intel(R) 8 Series/C220 Series USB EHCI #2 - 8C2D
IRQ 16	Intel(R) Management Engine Interface
IRQ 19	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
IRQ 19	Intel(R) Active Management Technology - SOL (COM3)
IRQ 22	High Definition Audio Controller
IRQ 23	Intel(R) 8 Series/C220 Series USB EHCI #1 - 8C26

C. Digital I/O Sample Code

File of the NCT5523D.H

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H                1
//---------------------------------------------------------------------------
#define     NCT5523D_INDEX_PORT         (NCT5523D_BASE)
#define     NCT5523D_DATA_PORT          (NCT5523D_BASE+1)
//---------------------------------------------------------------------------
#define     NCT5523D_REG_LD             0x07
//---------------------------------------------------------------------------
#define NCT5523D_UNLOCK                 0x87
#define     NCT5523D_LOCK               0xAA
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif//__NCT5523D_H
```

File of the MAIN.CPP

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//---------------------------------------------------------------------------
int main (void);

void Dio5Initial(void);
void Dio5SetOutput(unsigned char);
unsigned char Dio5GetInput(void);
void Dio5SetDirection(unsigned char);
unsigned char Dio5GetDirection(void);
//---------------------------------------------------------------------------
int main (void)
{
	char SIO;

	SIO = Init_NCT5523D();
	if (SIO == 0)
	{
		printf("Can not detect Nuvoton NCT5523D, program abort.\n");
		return(1);
	}

	Dio5Initial();

	//for GPIO20..27
	Dio5SetDirection(0x0F); //GP20..23 = input, GP24..27=output
	printf("Current DIO direction = 0x%X\n", Dio5GetDirection());

	printf("Current DIO status = 0x%X\n", Dio5GetInput());

	printf("Set DIO output to high\n");
	Dio5SetOutput(0x0F);

	printf("Set DIO output to low\n");
	Dio5SetOutput(0x00);

	return 0;
}
//---------------------------------------------------------------------------
```

```
void Dio5Initial(void)
{
        unsigned char ucBuf;

    ucBuf = Get_NCT5523D_Reg(0x1C);
    ucBuf &= ~0x02;
    Set_NCT5523D_Reg(0x1C, ucBuf);

        Set_NCT5523D_LD(0x07);                          //switch to logic device 7
        //enable the GP2 group
        ucBuf = Get_NCT5523D_Reg(0x30);
        ucBuf |= 0x04;
        Set_NCT5523D_Reg(0x30, ucBuf);
}
//---------------------------------------------------------------------------
void Dio5SetOutput(unsigned char NewData)
{
        Set_NCT5523D_LD(0x07);                      //switch to logic device 7
        Set_NCT5523D_Reg(0xE1, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
        unsigned char result;

        Set_NCT5523D_LD(0x07);                      //switch to logic device 7
        result = Get_NCT5523D_Reg(0xE1);
        return (result);
}
//---------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
        //NewData : 1 for input, 0 for output
        Set_NCT5523D_LD(0x07);                      //switch to logic device 7
        Set_NCT5523D_Reg(0xE8, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
        unsigned char result;

        Set_NCT5523D_LD(0x07);                      //switch to logic device 7
        result = Get_NCT5523D_Reg(0xE8);
        return (result);
}
//---------------------------------------------------------------------------
```

File of the NCT5523D.CPP

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
	unsigned int result;
	unsigned char ucDid;

	NCT5523D_BASE = 0x4E;
	result = NCT5523D_BASE;

	ucDid = Get_NCT5523D_Reg(0x20);
	if (ucDid == 0xC4)					//NCT5523D??
	{	goto Init_Finish;	}

	NCT5523D_BASE = 0x2E;
	result = NCT5523D_BASE;

	ucDid = Get_NCT5523D_Reg(0x20);
	if (ucDid == 0xC4)					//NCT5523D??
	{	goto Init_Finish;	}

	NCT5523D_BASE = 0x00;
	result = NCT5523D_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
	outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
	outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
	outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//---------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
        outportb(NCT5523D_DATA_PORT, LD);
        Lock_NCT5523D();
}
//---------------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        outportb(NCT5523D_DATA_PORT, DATA);
        Lock_NCT5523D();
}
//---------------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        Result = inportb(NCT5523D_DATA_PORT);
        Lock_NCT5523D();
        return Result;
}
//---------------------------------------------------------------------------
```

D. Watchdog Timer Configuration

The WDT is used to generate a variety of output signals after a user programmable count. The WDT is suitable for use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sorts of circumstances, the timer will count to zero and the selected outputs will be driven. Under normal circumstance, the user will restart the WDT at regular intervals before the timer counts to zero.

SAMPLE CODE:

```
File of the NCT5523D.H
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H            1
//---------------------------------------------------------------------------
#define     NCT5523D_INDEX_PORT         (NCT5523D_BASE)
#define     NCT5523D_DATA_PORT          (NCT5523D_BASE+1)
//---------------------------------------------------------------------------
#define     NCT5523D_REG_LD             0x07
//---------------------------------------------------------------------------
#define NCT5523D_UNLOCK             0x87
#define     NCT5523D_LOCK               0xAA
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif//__NCT5523D_H
```

File of the MAIN.CPP.

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//---------------------------------------------------------------------------
int main (void);

void WDTInitial(void);
void WDTEnable(unsigned char);
void WDTDisable(void);

//---------------------------------------------------------------------------
int main (void)
{
	char SIO;

	SIO = Init_NCT5523D();
	if (SIO == 0)
	{
		printf("Can not detect Nuvoton NCT5523D, program abort.\n");
		return(1);
	}

	WDTInitial();

	WDTEnable(10);

	WDTDisable();

	return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
	unsigned char bBuf;
	Set_NCT5523D_LD(0x08);					//switch to logic device 8
	bBuf = Get_NCT5523D_Reg(0x30);
	bBuf &= (~0x01);
	Set_NCT5523D_Reg(0x30, bBuf);				//Enable WDTO
}
//---------------------------------------------------------------------------
```

```
void WDTEnable(unsigned char NewInterval)
{
        unsigned char bBuf;

        Set_NCT5523D_LD(0x08);                          //switch to logic device 8
        Set_NCT5523D_Reg(0x30, 0x01);                   //enable timer

        bBuf = Get_NCT5523D_Reg(0xF0);
        bBuf &= (~0x08);
        Set_NCT5523D_Reg(0xF0, bBuf);                   //count mode is second

        Set_NCT5523D_Reg(0xF1, NewInterval);        //set timer
}
//---------------------------------------------------------------------------
void WDTDisable(void)
{
        Set_NCT5523D_LD(0x08);                          //switch to logic device 8
        Set_NCT5523D_Reg(0xF1, 0x00);                   //clear watchdog timer
        Set_NCT5523D_Reg(0x30, 0x00);                   //watchdog disabled
}
//---------------------------------------------------------------------------
```

File of the NCT5523D.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
	unsigned int result;
	unsigned char ucDid;

	NCT5523D_BASE = 0x4E;
	result = NCT5523D_BASE;

	ucDid = Get_NCT5523D_Reg(0x20);
	if (ucDid == 0xC4)						//NCT5523D??
	{	goto Init_Finish;	}

	NCT5523D_BASE = 0x2E;
	result = NCT5523D_BASE;

	ucDid = Get_NCT5523D_Reg(0x20);
	if (ucDid == 0xC4)						//NCT5523D??
	{	goto Init_Finish;	}

	NCT5523D_BASE = 0x00;
	result = NCT5523D_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
	outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
	outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
	outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//---------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
        outportb(NCT5523D_DATA_PORT, LD);
        Lock_NCT5523D();
}
//-------------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        outportb(NCT5523D_DATA_PORT, DATA);
        Lock_NCT5523D();
}
//-------------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        Result = inportb(NCT5523D_DATA_PORT);
        Lock_NCT5523D();
        return Result;
}
//-----------------------------------------------------------------------
```